Load Balancing in Computational Grids Using Ant Colony Optimization Algorithm

[1]SOWMYA SURYADEVERA, [2]JAISHRI CHOURASIA, [3]SONAM RATHORE & [4]ABDUL JHUMMARWALA

[1,2&3]MITS,Lakshmangarh, [4]Bisag,Gandhinagar
E-mail : sowmya.surya19@gmail.com[1], jaishri.chourasia14@gmail.com[2], sonamrathore18@gmail.com[3], abdul.zummerwala@yahoo.co.in[4]

Abstract – Grid computing is the combination of computer resources from multiple administrative domains for a common goal. Load balancing is one of the critical issues that must be considered in managing a grid computing environment. It is complicated due to the distributed and heterogeneous nature of the resources. An Ant Colony Optimization algorithm for load balancing in grid computing is proposed which will determine the best resource to be allocated to the jobs, based on resource capacity and at the same time balance the load of entire resources on grid. The main objective is to achieve high throughput and thus increase the performance in grid environment.

Keywords *- Grid computing, Ant colony optimization, Grid load balancing, performance, Grid, ANT algorithm, Scheduling.*

I. INTRODUCTION

Grid computing offers seamless access to rare and limited resources. Grid computing (or the use of a computational grid) is applying the resources of many computers in a network to a single problem at the same time usually to solve a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. A computational grid is the cooperation of distributed computer systems where user jobs can be executed on local or remote computer systems. On one side it provides the user with access to locally unavailable resource types on the other hand there is the expectation that a larger number of resources are available. A data grid denotes systems that provide a hardware and software infrastructure for synthesizing new information from data repositories that are distributed in a wide area network. Load balancing [1] and resource scheduling algorithm plays important role in the resource management and also gives much impact in the performance point of view.

Scheduling is process that maps and manages the execution of inter- dependent tasks on the distributed resources. The demand for scheduling is to achieve high performance computing. It aims to find a suitable allocation of resources for each job. It allocates suitable resources to tasks so that the execution can be completed to satisfy objective functions imposed by users. In Grid environments, scheduling decisions must be made in the shortest time possible, because there are many users competing for resources, and time slots desired by one user could be taken up by another user at any moment.

The load balancing mechanism aims to equally spread the load on each computing node, maximizing their utilization and minimizing the total task execution time. Load balancing is a mechanism which equally spreads the load. It minimizes the response time and improves the resource utilization rate. Load balancing in grid can be done based on the conditions of the system such as static or dynamic [7].

Static load balancing:

In static load balancing, algorithms scheduling is based on a default policy. Condition of system in every scheduling will be specified and on that base scheduling takes place and no more scheduling will take place until the work is done.

Dynamic load balancing:

A dynamic load balancing algorithm adapts its decision with the system that means processing duties with changing system condition. Dynamic load balancing makes decision on basis of the present system condition and quickly adapts with workload fluctuations.

This paper presents algorithm based on the general ant adaptive scheduling heuristics and an added in load balancing guide component. Section 2 describes the use variants of ant colony optimization

(ACO) algorithm in grid computing. The proposed algorithm is discussed in Section 3.The concluding remarks are highlighted in Section 4.

II. ACO ALGORITHMS IN GRID

ACO [1] is inspired by a colony of ants that work together in foraging behaviour. This behaviour encouraged ants to find the shortest path between their nest and food source. Every ant will deposit a chemical substance called pheromone on the ground after they move from the nest to food sources and vice versa. Therefore, they will choose an optimal path based on the pheromone value. The path with high pheromone value is shorter than the path with low pheromone value. This behaviour is the basis for a cooperative communication. There are various types of ACO algorithm such as Ant Colony System (ACS), Max- Min Ant System (MMAS), Rank-Based Ant System (RAS) and Elitist Ant System (EAS) [5].

ACO has been applied in solving many problems in scheduling such as Job Shop Problem, Open Shop Problem, Permutation Flow Shop Problem, Single Machine Total Tardiness Problem, Single Machine Total Weighted Tardiness Problem, Resource Constraints Project Scheduling Problem, Group Shop Problem and Single Machine Total Tardiness Problem with Sequence Dependent Setup Times [5]. A recent approach of ACO researches in the use of ACO for scheduling job in grid computing [6]. ACO algorithm is used in grid computing because it is easily adapted to solve both static and dynamic combinatorial optimization problems. In [2], ACO has been used as an effective algorithm in solving the load balancing problem in grid computing. The process taken by ACO will consider the pheromone value which depends on the time taken by each resource to process jobs. It does not consider the capacity of resources such as their bandwidth, processor speed and load.

In [4], two distributed artificial life-inspired load balancing algorithm are introduced, which are ACO and Particle Swarm Optimization (PSO). In the proposed algorithm, an ant acts as a broker to find the best node in term of the pheromone value stored in the pheromone table. The node with the lightest load is selected as the best node. The position of each node in the flock can be determined by its load in PSO. The particle will compare the load of nodes with its neighbours and will move towards the best neighbour by sending assigned jobs to it. The proposed algorithm performed better than ACO for job scheduling where jobs are being submitted from different sources and different time intervals. However, PSO uses more bandwidth and communication compared to ACO.

A study in [3] proposed a new algorithm that is based on an echo intelligent system, autonomous and cooperative ants. In this proposed algorithm, the ants can procreate and also can commit suicide depending on existing condition. Ant level load balancing is proposed to improve the performance of the mechanism. Ants are created on demand during their lives adaptively to achieve the grid load balancing. The ants may bear offspring when they detect the system is drastically unbalanced and commit suicide when they detect equilibrium in the environment. The ants will care for every node visited during their steps and record node specifications for future decision making. Theoretical and simulation results indicate that this new algorithm surpasses its predecessor. However, the pheromone values were not updated in this proposed algorithm which enables the assignment of jobs to the same resource.

ACO algorithm for load balancing in distributed systems through the use of multiple ant colonies is proposed in [8]. In this algorithm, information on resources is dynamically updated at each ant movement. Load balancing system is based on multiple ant colonies information. Multiple ant colonies have been adopted such that each node will send a coloured colony throughout the network. Coloured ant colonies are used to prevent ants of the same nest from following the same route and also enforcing them to be distributed all over the nodes in the system and each ant acts like a mobile agent which carries newly updated load balancing information to the next nodes. This proposed algorithm has been compared with the work-stealing approach for load balancing in grid computing. Experimental result shows that multiple ant colonies work better than work-stealing algorithm in term of their efficiency. However, the multiple ant colonies do not consider resources capacity and jobs characteristics. This can make matching the jobs with the best resources a difficult task for the scheduling algorithm.

An enhanced ant algorithm for task scheduling in grid computing was proposed in [9], which gives better throughput with a controlled cost. The proposed scheduling algorithm increased the performance in terms of low processing time and low processing cost when applied to a grid application with a large number of jobs such as parameter sweeps application. This algorithm works effectively in minimizing the processing time and processing cost of the jobs. The simulation results of various scheduling algorithm such as modified ant algorithm and cost controlled algorithm are also compared. The result shows that this enhanced algorithm works better than the ant algorithm. By considering the processing cost, this enhanced ant algorithm is more suitable for wide use. However, this algorithm does not consider the size of the jobs which leads to inappropriate assignment of jobs to resources.

Based on the previous research discussed above, there have been some or the other problems while

scheduling or balancing load in a grid environment. Taking some of the problems into consideration ant colony optimization algorithm for load balancing in grid computing is proposed which will determine the best resource to be allocated to the jobs, based on resource capacity and at the same time balance the load of entire resources on grid.

III. PROPOSED ANT ALGORITHM FOR GRID LOAD BALANCING

When a resource j enrolls into the grid system, it is asked to submit its performance parameters, such as the number of processor, processing capability MIPS (Mega Instruction per Second) of each processor, its RAM capability and the communication ability etc. Based on these parameters the initial pheromone value i.e. trail intensity for resource j is calculated as:

$$\tau j(0) = n \times p + Sj + M \quad (1)$$

where n is the No. of processors, p is MIPS of each processor, the parameter Sjis the communication bandwidth ability and M is the RAM capability of resource j.

The trail intensity at time t is updated and given by:

$$\tau j(t) = \rho \times \tau j(t-1) + \Delta\tau j(t-1,t) \quad (2)$$

Where, $\Delta\tau j(t-1,t)$ is the variety of quantity of the trail substance laid on the path from the scheduling center to resource j between time t-1 and t; ρ is the permanence of pheromone ($0 < \rho < 1$); (1-ρ) is an evaporation of pheromone.

When job is assigned to resource j the pheromone value of that resource is updated such that $\Delta\tau j(t-1,t) = -k$, where k is the coefficient relevant to computation workload and communication quantity of the job.

When a job is successfully completed and the resource j is released then there would be an increment in the pheromone value of that resource, $\Delta\tau j(t-1,t) = Ce \times k$, where Ce is the encouragement coefficient.

When a job fails and the resource j is released then there would be decrement in the pheromone value of the resource, $\Delta\tau j(t-1,t) = Cp \times k$, where Cp is the punishment coefficient.

Different chosen values of the above mentioned coefficients ρ, k, Ce, Cp will change the value of $\Delta\tau j(t-1,t)$, and when to update $\tau j(t)$ cause different instantiation of the ant algorithm.

The possibility of task assignment to every resource will be recomputed as:

$$\rho_j(t) = \frac{[\tau_j(t)]^{\alpha} * [\eta_j(t)]^{\beta}}{\sum_u [\tau_u(t)]^{\alpha} * [\eta_u]^{\beta}} \quad (3)$$

where $\tau j(t)$ the trail intensity on the path form scheduling center to resource j at time t; ηj is called visibility, namely the innate performance quantity of the resource j ($\tau j(0)$); α is the parameter on the relative importance of trail intensity; β is the parameter on the relative importance of visibility.

The jobs are allocated to resources based on the pheromone value calculated by considering resource parameters. But by the above method if a particular resource always completes jobs successfully then the resource gets loaded heavily. This results in the bottleneck in the grid and influences to completing the jobs. Therefore we introduce load balancing factor in the ant algorithm to improve the load balancing capability.

We thus introduce the load balancing factor λjof resource j, which would depend on the job finishing rate of the resource j. The load balancing factor will change the trial intensity from $\Delta\tau j$to $\Delta\tau j + C\ \lambda j$ ($C > 0$ is a coefficient of the load balancing factor), the more jobs finished the more increases the trial intensity,contrarily, the more jobs not completed, the more decreases the trail intensity. So by introducing the load balancing factor the load on all the resources in the grid will be balanced.

IV. CONCLUSION AND FUTURE WORK

The proposed algorithm is expected to determine the best resource to be allocated to the jobs, based on resource capacity and at the same time to balance the load in grid. The algorithm considers various resource parameters (MIPS, communication bandwidth, No. of processors and memory) for calculating pheromone i.e. the resource capability for executing various jobs thus allocating the best resource for the job and at the same time balance the load of all the resources. The algorithm is expected to achieve better throughput and hence increase the overall performance in the grid environment. In the future the algorithm will be implemented using GridSim simulator.

VI. ACKNOWLEDGMENTS

We would like to thank Mr. P. K. Bishnoi of MITS, Lakshmangarh and Mr. M.B. Potdar of BISAG, Gandhinagar for their support and guidance.

REFERENCES

[1] Jagdish Chandra Patni, Dr. M.S.Aswal, Om Prakash Pal and Ashish Gupta, "Load balancing Strategies for Grid Computing" presented at 3rd international conference on electronics computer technology (ICECT), vol.3, pp. 239-243, 2011.

[2] Y. Li, "A Bio-inspired Adaptive Job Scheduling Mechanism on a Computational Grid," International Journal of Computer Scienceand Network Security (IJCSNS), vol. 6(3), pp. 1-7, 2006.

[3] M. Salehi and H. Deldari, "Grid load balancing using an echo system of intelligent ants," presented at Proceedings of the 24th IASTED international conference on Parallel and distributed computing and networks, 2006.

[4] A. Moallem, and S. A. Ludwig, "Using Artificial Life Techniques for Distributed Grid Job Scheduling" presented at ACM Symposium on Applied Computing (SAC 2009), Hawaii, U.S.A., pp. 1091 – 1097, 2009.

[5] M. Dorigo and T. Stützle, Ant colony optimization, Cambridge,Massachusetts, London, England: MIT Press, 2004.

[6] S. Fidanova and M. Durchova, "Ant algorithm for grid scheduling problem," Lecture Notes in Computer Science, vol. 3743, pp. 405- 412, 2006.

[7] Moradi, M.Dezfuli, M.A.Safavi, "A new time optimizing probabilistic load balancing algorithm in grid computing" presented at 2nd International Conference on Computer Engineering and Technology (ICCET), vol. 1, pp. v1232-v1237, 2010.

[8] A. Ali, M. A. Belal and M. B. Al-Zoubi, "Load Balancing of Distributed system Based on Multiple Ant Colonies Optimization,"American Journal of Applied Sciences, vol. 7(3), pp. 433-438, 2010.

[9] K. Sathish and A. Reddy, "Enhanced ANT Algorithm Based Load Balanced Task Scheduling in GRID Computing," IJCSNS, vol. 8, pp. 219, 2008.